NASA/TM—2008-215153

A Globally Optimal Particle Tracking Technique for Stereo Imaging Velocimetry Experiments

Mark McDowell
Glenn Research Center, Cleveland, Ohio

April 2008

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to *help@sti.nasa.gov*

- Fax your question to the NASA STI Help Desk at 301–621–0134

- Telephone the NASA STI Help Desk at 301–621–0390

- Write to:
  NASA Center for AeroSpace Information (CASI)
  7115 Standard Drive
  Hanover, MD 21076–1320

NASA/TM—2008-215153

A Globally Optimal Particle Tracking Technique for Stereo Imaging Velocimetry Experiments

Mark McDowell
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

April 2008

This work was sponsored by the Fundamental Aeronautics Program
at the NASA Glenn Research Center.

Level of Review: This material has been technically reviewed by technical management.

Available from

NASA Center for Aerospace Information
7115 Standard Drive
Hanover, MD 21076–1320

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161

Available electronically at http://gltrs.grc.nasa.gov

A Globally Optimal Particle Tracking Technique for Stereo Imaging Velocimetry Experiments

Mark McDowell
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Abstract

An important phase of any Stereo Imaging Velocimetry experiment is particle tracking. Particle tracking seeks to identify and characterize the motion of individual particles entrained in a fluid or air experiment. We analyze a cylindrical chamber filled with water and seeded with density-matched particles. In every four-frame sequence, we identify a particle track by assigning a unique track label for each camera image. The conventional approach to particle tracking is to use an exhaustive tree-search method utilizing greedy algorithms to reduce search times. However, these types of algorithms are not optimal due to a cascade effect of incorrect decisions upon adjacent tracks. We examine the use of a guided evolutionary neural net with simulated annealing to arrive at a globally optimal assignment of tracks. The net is 'guided' both by the minimization of the search space through the use of prior limiting assumptions about valid tracks and by a strategy which seeks to avoid high-energy intermediate states which can trap the net in a local minimum. A stochastic search algorithm is used in place of back-propagation of error to further reduce the chance of being trapped in an energy well. Global optimization is achieved by minimizing an objective function, which includes both track smoothness and particle-image utilization parameters. In this paper we describe our model and present our experimental results. We compare our results with a nonoptimizing, predictive tracker and obtain an average increase in valid track yield of 27 percent.

Introduction

Particle tracking is one of the phases of a typical Stereo Imaging Velocimetry (SIV) experiment that seeks to assign three-dimensional velocity vectors to the particles. Since camera images are two-dimensional, one can track in two-dimensions and then stereo-match tracks from left and right camera images to obtain three-dimensions (ref. 1). Alternatively, one can stereo-match the particles beforehand and track directly in three-dimensions (ref. 2). We have chosen the former approach because it reduces the problem search-space complexity by one-dimension while providing a step subsequent to the tracking phase that serves to prune invalid tracks. We will concentrate on a single camera view in this paper.

An *image frame* is an NTSC based digitized frame consisting of many particles. These particles appear as overlapping clusters in many areas and must be reduced into constituent particles and further reduced to geometrical points representing the approximate center of each particle. We apply a particle overlap decomposition routine followed by an intensity-weighted centroid determination algorithm (ref. 3). Each particle image may then be parameterized by the 3-tuple

$$\left[P_c(k,t), f_0 + t, L_c(k,t)\right] = \left[\left(x_c(k,t), y_c(k,t)\right), f_0 + t, \left(p_1, p_2, p_{3,\ldots}\right)_{k,t}\right] \tag{1}$$

where

f_0+t is the image frame sequence number,

$(x_c(k,t),y_c(k,t))$ is the particle centroid for particle image k in frame f_0+t, and $(p_1,p_2,p_3,\ldots)_{k,t}$ are the probabilities that a cluster consists of one, two, three, etc. constituent particles.

We take a uniform time sequence of images at 30 Hz. We label each particle in a given frame with a particle image id number. A *track* is then defined as a sequence of four of the 3-tuples shown in equation (1).

$$\left\{\begin{matrix}\left(P_c\left(k_1,1\right),f_0+1,L_c\left(k_1,1\right)\right),\left(P_c\left(k_2,2\right),f_0+2,L_c\left(k_2,2\right)\right),\\ \left(P_c\left(k_3,3\right),f_0+3,L_c\left(k_3,3\right)\right),\left(P_c\left(k_4,4\right),f_0+4,L_c\left(k_4,4\right)\right)\end{matrix}\right\} \tag{2}$$

It takes three such 3-tuples to define a constant acceleration, while the fourth frame is retained to provide a check on the validity of the constant acceleration assumption. Trigui, Guezennec, Brodkey, and Kent (ref. 1) track particles in a five frame sequence to enforce a more rigorous standard of track smoothness. However, longer sequences exclude more tracks in turbulent flow. They also increase the search space in an exponential manner (refs. 3 and 11). For these reasons, we elected to use four frames.

While much work has been done in the area of SIV particle tracking, the goal of attaining a globally optimal solution is a recent development. Miller et al. and Crouser et al. (refs. 3 and 5) introduced a Hopfield net model in this regard, but there are a few problems with this approach. One problem is that their Hopfield net employs a track-growing strategy that allows invalid and incomplete track fragments to form. Hopfield nets are prone to becoming stuck in local energy minima and suffer greatly from track-overlap errors, which form the basis for global optimization. During the intermediate stages of global optimization, tracks must pass through an overlap state of high activation energy that would serve to trap the net at a local minimum.

We developed a variant of a Guided Evolutionary Simulated Annealing neural net to perform global optimization. There are numerous advantages to our approach. First, this approach restricted the search space to complete, valid tracks, identified during a crucial setup phase. Second, the evolutionary net with simulated annealing is a stochastic search method that is less susceptible to being stuck in local minima. Third, an interventionist strategy was adopted whereby randomly selected tracks would not be laid down until any obstructing tracks were moved out of the way (or deselected altogether). This step, a simple acknowledgment of the fact that a valid configuration can never overlap, minimizes or eliminates high intermediate energy barriers. As mentioned previously, the number of images per time sequence per track was reduced from five to four, reducing computational complexity by an order of magnitude.

Tracking Methodology

Our algorithm divides the tracking problem into two parts, namely: initialization followed by optimization. Initialization creates a database containing all feasible tracks. In our neural net paradigm these tracks are equivalent to nodes. Optimization is a stochastic-search, evolutionary algorithm which begins by randomly selecting from the database complete sets of first guesses as to possible track assignments, called parents. Of course, these early parents are poor solutions and need to be optimized. This is done by randomly selecting one or more tracks within a parent, changing the track(s), and calling the resultant modified solution a child. Several children per parent are generated to create a family. Families evolve when a new generation of parents is chosen from among the existing family members. In general, the member of a given family with the lowest system error becomes the new parent, and the whole process is repeated over many generations until an acceptable solution is found or until the families stop evolving (the system error is the objective function which is to be minimized during optimization and will be explicitly defined in a later section). The latter can occur because the best solution was found, or, more likely, because a local minimum in the system error, also called an energy well, was found.

Terrain-following optimization algorithms such as back-propagation of error, tend to get stuck in local energy minima. Stochastic search methods are not trapped as easily and are more robust. Yip and Pao (ref. 6) devised the general approach that they called the Guided Evolutionary Simulated-Annealing optimization algorithm (GESA). A modified version is adapted for this paper.

Oftentimes, restricting the search space to only feasible solutions can result in increased performance. This is why the database chosen in this paper considers the entire track, consisting of four particle images, to constitute a single node in the net. Here, a node is a binary-decision variable representing selection or de-selection of a track option; and the net is the set of functional constraints existing between the nodes, and expressed as an objective function to be minimized.

The search space is further restricted by screening out invalid tracks during initialization. The ideal track is considered to be one that has bounded, constant acceleration across the sequence of four images. A maximum velocity restriction is also imposed. These constraints place bounds on where to search for particles belonging to a track in successive frames. These ideas are fully presented in a later section.

Evolution within families occurs by selecting the family member with the lowest system error. System error is a relative measure of the acceptability of a given state. Ideally, every particle in every frame should be assigned to a single track. This restriction is expressed in the system error (objective function) by a usage error term. Furthermore, the tracks so formed should each match exactly with a unique track in the orthogonal images of the second camera. This stereo matching requirement is not known *a priori* but serves as an after-the-fact measure of how successful the tracking algorithm may have been.

Particle utilization, however, by itself is clearly not enough information to determine correct tracks. The only other information available is track straightness and track smoothness. This is based on our assumption of constant acceleration. Tracks are biased both by how straight they are and by how well the fourth particle in the track conforms to the position predicted for it by a pair of parametric quadratic equations in the x and y directions. These equations are derived from the preceding three particles in that track. The equation for system error becomes:

$$\text{system\_error} = \text{track\_error} + \text{usageCoeff} * \text{particle\_usage\_error} \tag{3}$$

where

$$\text{track\_error} = \sum_{j=1}^{N_1} f\,(\text{track\_straightness\_error[j]}, \text{track\_smoothness\_error[j]}$$

N_1 = the number of tracks = the number of particles in frame 1

$f(\tau)$ = a penalty function defined as a mapping from the parameters shown, with $-2.0 < f(\ ) < 2.0$ (see fig. 7),

usageCoeff = weighting of the usage error relative to the track error,

$$\text{particle\_usage\_error} = \sum_{f=1}^{F}\sum_{j=1}^{N_f} \left|\ \text{assigned\_particle\_usage[f, j]} - 1\ \right|,$$

assigned_particle_usage[f, j] = the number of times particle j in frame f appears within any track from a particular assignment of tracks,

F = the number of frames in a track, and

N_f = the number of particles in frame f, $1 \le f \le F = 4$

Experimental Procedure

The experimental setup consists of two cameras oriented at approximately 90° to each other with respect to the experimental chamber to give the best spatial resolution (refs. 2 and 7). The chamber is a

transparent cylinder, 3 in. high by 3 in. internal diameter, mounted within a cube with 4 in. sides. The cube is transparent on the two sides facing the cameras and painted black on the other two sides. The bottom of the cube and the cylinder are transparent. A strobe light slaved to the camera sync generator illuminates the chamber from below. The cameras provide real-time, interlaced video image data at a refresh rate of 60 Hz. Only the even-line images are used, providing an effective refresh rate of 30 Hz. This is done to accommodate the relatively low velocities used in our experiments and to eliminate jitter caused by interlacing two images 1/60th of a second out of sync with respect to each other.

Both the cylinder and the cube are filled with water doped with a wetting agent and seeded with a known number of 300 μm diameter particles. Particle motion within the cylinder was achieved by using a stirrer bar at a constant speed such that a vortex did not occur. The images from the cameras were stored on a pair of laser videodisk recorders. These images are then preprocessed to eliminate background noise. Figure 1 shows a typical camera image with background noise removed.

Particle centroids are determined and overlapping particles are identified via the particle overlap decomposition stage (ref. 7), which uses a probabilistic algorithm to determine if a cluster consists of one, two or three overlapping particles with probabilities assigned as p_1, p_2, and p_3:

$$p_n = \text{the probability that the cluster contains n particles, } 1 \le n \le 3 \tag{4}$$

where $p_1+p_2+p_3 = 1$

The probabilities are passed on to the tracking phase to allow the optimization phase to calculate expected values for track usage error.

Track Initialization

After loading the image data (see eq. (1)), all valid track options are identified and stored. A *valid track* is defined as a 4-tuple of 3-tuples (see eq. (2)), beginning with frame 1 and ending with frame 4, which passes through a single particle image in each of the four frames under consideration, and meets the following constraints:

1. Frame 2 particle must lie within a pre-selected maximal radius of the track's frame 1 particle position (refs. 8 to 10). This limiting radius is chosen based upon an *a priori* estimate of maximal velocity for the experiment.
2. Frame 3 track element must satisfy three, velocity-dependent constraints, namely: a minimum velocity constraint, a maximum velocity constraint, and a maximum angle constraint (refs. 8 to 10). These constraints are functional mappings from the velocity derived between frames 1 and 2 and shown in figure 2.
3. Frame 4 track element must lie within a specified maximum radius of the position estimated for it based upon the information from frames 1 through 3. This constraint is actually the maximum deviation accepted from the assumption of constant acceleration for the ideal track (ref. 11).
4. A fixed limit is placed upon the number of frame particles that may be assigned to any given partial track ending in frame 3. This reduces the size of the search space from $O(P^4)$ to $O(P^3)$, where P is the average number of particles per frame, and can result in dramatic time savings for large problems. Selection is performed by sorting the frame 4 particle images by increasing distance from the estimated position derived in constraint 3, and picking those with the smallest distance which also meet the requirements in constraint 3. Figure 3 shows an example of some of these constraints in the determination of valid tracks for a sequence of four image frames.

This definition of valid tracks leads naturally to a multi-branched, four-level tree structure for valid-track storage, where successive levels of the tree correspond to successive image frames. Figure 4 gives an example of such a structure for a hypothetical case of five frame 1 particles. There is one such tree

structure for each particle image in frame 1. Each frame 1 particle image is the root of its own *valid-track tree.*

Track Selection

The track-optimization algorithm selects one valid-track tree at random from a given parent and alters the currently assigned track for that tree. Therefore, there is a need to quickly select a single tree from among the set of all frame 1 valid-track trees for that parent. Suppose there are N_1 such trees for each parent (N_1 is dependent only on the number of particles in frame 1, so it must be the same for all parents). We are interested in eventually testing all valid-track options per tree. In general, we want the probability of selection to be weighted by the number of leaves in the tree. We choose:

$$\gamma_1(k) = \frac{T_k}{\sum_{i=1}^{N_1} T_i} \tag{5}$$

where

γ_1 (k) is the probability of selecting the kth valid-track tree. The subscript '1' refers to the fact that we are selecting among particles at the frame 1 level.

T_i is the number of valid tracks (leaves) in the ith tree, $1 \le i \le N_1$

We would like to perform tree selection by choosing a single random number in the range 0 to 1. Therefore we need to convert γ into a distribution function:

$$\Gamma_1(k) = \sum_{j=1}^{k} T_j \Big/ \sum_{i=1}^{N_1} T_i \tag{6}$$

For example, **figure 4** depicts representative values for $\Gamma_1(k)$, labeling them as 'frame 1 selection distribution numbers.'

Once a particular valid-track tree has been selected, the optimization algorithm randomly chooses one of the valid-track options in the tree to become the assigned track for that tree. To perform such a selection for any given tree, using a single random number between 0 and 1, requires a separate and distinct distribution function for each tree. Because there are an exponentially large number of unique combinations of track assignments as solutions to the tracking problem, it is not possible to directly test all of them. Instead, we must favor the best options at the expense of the others, without totally suppressing the latter. Hence, we must weight the leaves of a given tree according to which is most likely to be the correct choice. Equation (3) defines the correctness of an outcome in terms of the track-error and the track-usage-error. We now describe a weighting of the leaves of a tree based upon these parameters. We will use the phrase '*track-fragment through frame 3*' to mean the first three particle id numbers of some four-frame track (or of some set of four-frame tracks having the same first three particle id numbers).

Whole-track options in a valid-track tree may be represented by the leaves of the tree arranged in a rectangular grid, as depicted in figure 5. A given row of the grid represents all of the frame 4 options for a given 'track-fragment through frame 3.' The best frame 4 option for a given 'track-fragment through frame 3' is placed in column 1 of the grid, the second best in column 2, and so on. Notice that we restrict the number of frame 4 options for each 'track-fragment through frame 3' to a constant value (5 in this case) in conformity with the fourth geometric constraint. Let us define an error function as follows:

u(k, r, c) = g(track_straightness_error[*k*,*r*,*c*], track_smoothness_error[k,r,c]) (7)
= error function for the k[th] valid track tree having indices,
(r, c) = (row, column) and 0 < u(k,r,c) < 1

We do not want to reward poor performance (high error), so we must complement the sense of error into a sense of reward, as follows:

$$v(k,r,c) = 1 - u(k,r,c) \tag{8}$$

Normalizing these values gives:

$$\gamma_4(k,r,c) = v(k,r,c) \Big/ \sum_{i=1}^{F_k} \sum_{j=1}^{F_{k,r}} v(k,i,j), \quad 0 \le \gamma_4(k,r,c) \le 1 \tag{9}$$

where

γ_4 (k,r,c) = the discrete probability density function for selecting a given track in the tree. The subscript '4' indicates that selection is occurring at the fourth level of the tree (the leaves of the tree) and that, therefore, whole tracks are being selected.
F_k = the number of 'track-fragments through frame 3' in the k[th] valid-track tree
$F_{k,\gamma}$ = the number of track options in row r of the k[th] track tree, $1 \le F_{k,r} \le 5$

Finally, to perform a selection using a single random number between 0 and 1, we require the probability distribution function of γ_4 :

$$\Gamma_4(k,r,c) = \sum_{j=1}^{r-1} \sum_{i=1}^{F_{k,j}} \gamma_4(k,j,i) + \sum_{i=1}^{c} \gamma_4(k,r,i) \tag{10}$$

Track Error

Equation (10) is based upon $u(k,r,c)$ in equation (7), which describes the error ascribed to a valid track due to its deviation from an ideal, perfectly smooth track. We now derive this error. Throughout this section we retain the same notation as in the preceding section. That is, all valid tracks for a given frame 1 particle form a tree as in figure 4, and the leaves of the tree are ordered by row and column in a grid as in figure 5.

Refering to equation (7), if we assume that the parameters of g( ) are also normalized, we can write:

u(k, r, c) = trkCoeff * track_straightness_error[k,r,c]
+ (1.0 - trkCoeff) * track_smoothness_error[k,r,c] (11)

where 0 < trkCoeff < 1 is a simple weighting.

The variable track_straightness_error[k, r, c] is the distance of the track in image frame 3 from an estimated position based upon an assumption of constant velocity derived between image frames 1 and 2. This distance is depicted in figure 6(a), where it is labeled delta3. We write:

$$\text{track\_straightness\_error}[k, r, c] = \text{straightCoeff} * \text{delta3}[k, r, c] \quad (12)$$

where

delta3[k, r, c] = the euclidean distance between the constant velocity estimate of position, est3[k,r,c], and the actual position, id3[k,r,c], in frame 3 for the valid track in row r and column c of the valid-track grid for the k^{th} tree.

est3[k, r, c] = the estimated point with coordinates:

est3[k,r,c](x) = 2 * (id2[k,r,c](x) - id1[k,r,c](x)) + id1[k,r,c](x)

est3[k,r,c](y) = 2 * (id2[k,r,c](y) - id1[k,r,c](y)) + id1[k,r,c](y)

straightCoeff = a velocity-mapped term which makes the error independent of scale.

In figure 6(a)*, minR, maxR, Radius,* θ*, and max*θ are the velocity-mapped constraints on the size of the search region in image frame 3, mentioned earlier when discussing the geometrical constraints.

Similarly**,** track_smoothness_error[k, r, c] is the distance of the track in image frame 4 from an estimated position based upon an assumption of constant acceleration derived from the first three image frames. This distance is defined as delta4 in figure 6(b). Delta4 is the distance of the actual frame 4 particle position from the estimate est4. Est4 is arrived at by assuming constant acceleration as determined by the first three frames and expressed as a pair of quadratic, parametric equations in the X and Y dimensions. We have:

$$\text{track\_smoothness\_error}[k, r, c] = \text{smoothCoeff} * \text{delta4}[k, r, c] \quad (13)$$

where

delta4[k, r, c] = the euclidean distance between the constant-acceleration estimate of position, est4[k,r,c], and the actual position, id4[k,r,c], in frame 4 for the valid track in row r and column c of the k^{th} track tree.

est4[k, r, c] = the estimated point with coordinates derived below.

smoothCoeff = a velocity-mapped term which makes the error independent of scale.

and

$$\begin{aligned} \text{est4}[k, r, c](x) &= at^2 + bt + c = 9a + 3b + c \\ &= 3(\text{id3}[k,r,c](x) - \text{id2}[k,r,c](x)) + \text{id1}[k,r,c](x) \\ \text{est4}[k, r, c](y) &= a_2t^2 + b_2t + c_2 = 9a_2 + 3b_2 + c_2 \\ &= 3(\text{id3}[k,r,c](y) - \text{id2}[k,r,c](2)) + \text{id1}[k,r,c](y) \end{aligned}$$

Having defined u[k, r, c], we are able to derive equation (10), which allows us to randomly designate one of the tracks in the k^{th} track-tree as an active track. We denote these errors as being specific to a given parent (or child), labeled *p***,** as follows:

$$\text{system\_error}[p] = \text{track\_error}[p] + \text{usageCoeff} * \text{particle\_usage\_error}[p] \quad (14)$$

where

$$\text{track\_error}[p] = \sum_{k=1}^{N_1} \sum_{r=1}^{F_k} \sum_{c=1}^{F_{k,r}} f\,(u[p, k, r, c]) * X[p, k, r, c]$$

f(u[p, k, r, c]) = a penalty function for parent ‘p’ defined as a mapping of error

u[p, k, r, c] into the range $-2.0 < f(\,) < 2.0$ (see fig. 7)

X[p, k, r, c] = a binary variable with value 1 if the valid-track option [p, k, r, c] is active (has been selected), and with value 0 otherwise

N_1 = the number of track trees = the number of particles in frame 1

F_k = the number of 'track-fragments through frame 3' in the k^{th} valid-track tree

$F_{k,r}$ = the number of track options in frame 4 corresponding to the r^{th} 'track-fragment through frame 3' of the k^{th} track tree, $1 \le F_{k,r} \le 5$

The mapping f:[0, 1] $\rightarrow$ [-2, 2] is depicted in figure 7. The positive values of $f(\tau)$ in figure 7 reflect a preference to reject a certain track (since the optimization algorithm seeks to minimize the objective function). A zero value would show no preference between accepting a track or having no track at all selected in a given tree. Finally, a negative value of $f(\tau)$ induces track selection. The curve in figure 7 is skewed downwards to favor the negative range of $f(\tau)$. This allows a wider range of tracks to be selected. Therefore, tracks with higher track error are made more acceptable. We would prefer to have track usage-errors determine the final outcome because they are the single source of global interaction during optimization. We thus look upon the track smoothness constraint as a local bias favoring smoother tracks, applied against a larger backdrop of global optimization on track usage errors.

Track Optimization

Recall that evolutionary optimization places distinct copies of the entire problem space (the valid-track trees and any track assignments made therein) in each of the parents. Then the parents create clones of themselves, called children, which can subsequently be modified to create a new generation of track assignments. The relative performance of the children is evaluated by calculating the system error defined in equations (3) and (14). The child with the lowest system error can become the new parent.

Particle Image Usage Error

Recall the definition of particle_usage_error given in equation (3), assigning them to a particular parent, labeled p:

$$\text{particle\_usage\_error}[p] = \sum_{f=1}^{F} \sum_{j=1}^{N_f} \left| \text{assigned\_particle\_usage}[p, f, j] - 1 \right| \qquad (15)$$

$$\text{assigned\_particle\_usage}[p, f, j] = \sum_{k=1}^{N_1} \sum_{r=1}^{F_k} \sum_{c=1}^{F_{k,r}} X[p, k, r, c] * Y[p, k, r, c, f, j]$$

= the number of times particle j in image frame f appears within any track from a particular assignment of tracks for parent p.

X[p, k, r, c] = a binary variable with value 1 if the valid-track option [p, k, r, c] is active (has been selected), and with value 0 otherwise

Y[p, k, r, c, f, j] = a binary variable with value 1 if the valid-track option [p, k, r, c] contains particle j in frame f , and with value 0 otherwise

F = the number of frames in a track = 4

N_f = the number of particles in frame f, $1 \le f \le 4$

Notice that equation (15) says that if assigned_particle_usage[] equals unity then there is no error. This is equivalent to saying that every particle should be in only one track. Error increases linearly as assigned_particle_usage[] deviates from unity from above or below.

Child Implementation

Recall that optimization of the parent involves the generation of children which are clones of the parent subsequently altered with respect to the assignment of one (or a couple) track(s). An implementation strategy is to record only the changes for the children, as opposed to making explicit copies of all of the track-trees in the problem. Child system errors can be calculated by altering the parent to look like the child, performing the calculations, them changing it back. The savings in memory and time required can be substantial for large problems.

The Optimization Process

The algorithm for executing the optimization routine is shown in figure 8. The main loop runs the search until a stable state is found. Typically this occurs in 60 to 160 iterations for 100 particles or up to 500 iterations for 200 to 300 particles. The 'for' loop makes changes to all parents. We discuss aspects of this procedure in the next several sections.

Child Creation

A child is created from its parent when a track tree is randomly chosen for alteration, using equation (6). A specific track in the tree is chosen using equation (10). The chosen track is to be made active in the sense of the binary variables X[ ] and Y[ ] in equations (14) and (15) respectively.

Parent Selection

Once the children are created, the optimization procedure described in figure 8 examines the outcomes and determines how successful the different families have been. The measure for such determination is equation (3), the system error for parent or child. The first task in this routine is to find the number of viable children in each family. A child is viable if its system error is at least as good as its parent's system error. Those children that do not meet this criterion still have a chance to be rated viable if they overcome a probabilistic hurdle known as stochastic simulated annealing, which is similar to the thermodynamic annealing factor found in Boltzmann machines. The test is as follows:

$$\text{if } (\exp(\text{delta\_error} / \tau) > \xi \text{) then child is valid} \quad \text{else child is invalid} \tag{16}$$

where

delta_error = (parent_error - child_error) < 0
τ = thermodynamic annealing factor, the annealing 'temperature'
ξ = randomly chosen real number, with $0 < \xi < 1.0$

The purpose of this test is to decrease the probability that the optimization will get stuck in a local energy minimum by allowing some borderline children to be selected while still maintaining an overall push in the direction of the steepest decline in the system error terrain.

In this manner, each parent, p, receives a valid child count that is a relative measure of the success of a family. Each family is assigned a new family size based upon this count as follows:

$$\text{childCount}[p] = \text{total\_count} * \text{valid\_count}[p] \Big/ \sum_{p=1}^{P} \text{valid\_count}[p] \quad (17)$$

where

total_count = the total desired population of children

After family sizes are determined, the next task is to designate the new parent for the next iteration. This is called reseeding the families. It is necessary to assign the 'best' family by the process of comparing system errors and picking the family with the smallest error.

Reseeding the Families

We give families a chance to perform by assigning them a number of '*lives*' that are initialized to some value, say 5. Each family is thus given 5 lives in which to perform. Each time a family fails to install a new parent, one life is lost. Each time it succeeds, a new life is added (up to the limit). This becomes a race for the family to keep producing. If a family runs out of lives, then it is replaced by one of the viable children of the best family. Once assigned, that child is marked unavailable for further assignment. If the best family has no such children left to assign, then the best parent is assigned.

The assignment of lives to each family is a useful regulatory mechanism for controlling the life span of each family. It gives marginal families a chance to overcome local minima while concentrating efforts around the best families.

Using Stereo-Matched Tracks for Verification

Performance is measured by the percent of valid stereo-matched tracks obtained. The number of actual tracks present was measured by simply counting the number of particle images in each frame. Separately, an optimizing stereo-matching program was used to provide a list of valid stereo-matched tracks.

The stereo-matching algorithm used for validation purposes is an evolutionary neural net model, conceptually similar to the one used for the tracking model. Optimization error is taken to be the sum of the squared vertical displacements (in pixels) between corresponding particle images, on a frame-by-frame basis, matching a given track from the left view with one in the right view. This is similar to the approach taken in Trigui et al. (ref. 1). A threshold value was chosen for the sum of the squared errors. Any pair of tracks having an error exceeding this value was rejected.

We chose a threshold value of 4 * (2 * 2) = 16 which is what one gets if each of the four frames has a vertical alignment error of 2 pixels between the left and right views. We choose the error to be the sum of the squared errors in each frame to prevent the displacement of any single particle in the tracks from being excessive. We also are able to directly examine the tracks generated using a stylized representation of the tracks generated, and of failed tracks to provide a means to visually verify that the tracks are correct.

Experimental Results

The track yields of the optimizing and the non-optimizing models are shown in figure 9 for a run whose density were 130 particles. Here we have taken a sequence of 16 frames and have successively chosen four frames whose starting frame varies from frame 1 through frame 13 in succession. The average yields are 86 and 107 valid tracks for the non-optimizing and the optimizing models,

respectively. The average yields for runs of varying densities are shown in table 1. The average track yield for the non-optimizing model is 66 percent, while for the optimizing model it is 84 percent. From the final column of table 1, we see that, *on average, optimization in the tracking phase alone brought about a 27 percent increase in yield.*

Obviously, track yields are highest at low particle densities due to less interaction between tracks. Also, the final column shows that the relative advantage of the optimizing model over the non-optimizing model was greater at higher densities. This is expected due to the increased overlap between tracks and the corresponding need to distinguish between a greater number of options per track. Also, as overlap increases, the effect of track assignments in one region of space begins to influence track assignments in more remote areas through a domino effect of one assignment pushing out another. This calls for an increasingly global approach to optimization through track utilization error minimization.

The second column of table 1 shows that the area covered by the particles at the densities listed varies from 1.3 up to 4.9 percent of the total area, which is a significant increase (Malik et al. provide a much more thorough discussion of the measurement of particle density and its effects upon tracking yields). We expected to see a significant drop-off in track yield as the number of particles reached 270.

In the case of the optimizing model, the reason for track failure is generally the lack of sufficient information in each camera view to resolve intertwining tracks. Figure 10 depicts some of the situations where ambiguity can arise. In all cases depicted in figure 10, the incorrectly assigned track is valid in the sense that it is smooth. The laminar flow example in figure 10 is interesting because of its intrusiveness. Usually when such tracks form they do not prevent the other tracks from forming. However, such cases have high overlap error. This is not completely avoidable and happens quite a bit in tight clusters of particles. Errors like those depicted in table 1 are typical of the errors which cause a drop in performance at higher particle densities (from 88 percent for 90 particles to 82 percent for 270 particles).
All of the examples in figure 10 are reasons why yields drop dramatically as density increases.

Summary

We have shown significant improvement in valid, stereo-matched track yields through the use of global optimization techniques during the tracking phase. This was done using short time-sequences of only four time frames. We have shown that virtually all tracks can be readily characterized as belonging to one of two classes characterized by the degree of track integrity, and we have shown how to adjust tracks of low track integrity to provide a more accurate estimate of track shape.

The tracking program described in this paper utilizes the probabilities output from a probabilistic particle overlap-decomposition algorithm in order to more fully utilize the available information.

The optimizations converge very quickly due to the careful efforts to reduce the number of potentially valid tracks during initialization, and due to the efforts taken by the child-creation procedure to not create unstable states by blindly laying new tracks over old. The reduction in the number of potentially valid tracks occurs via a set of geometrical constraints that serve to restrict the size of the search space along the length of the track.

References

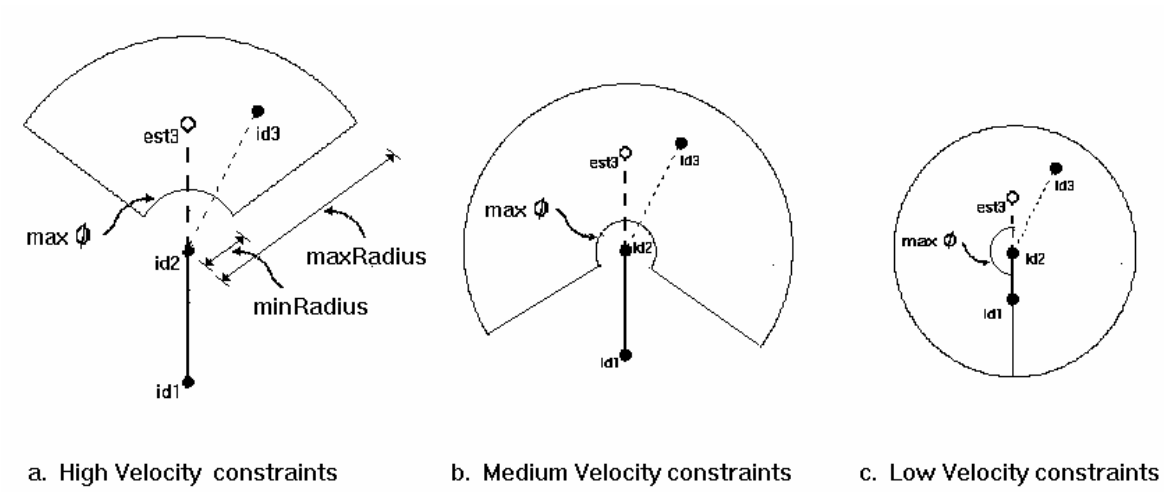
1. N. Trigui, Y. Guezennec, R. Brodkey, and C. Kent, “Fully Automated Three-Dimensional Particle Image Velocimetry System Applied to Engine Fluid Mechanics Research,” Proc. of Optical Methods and Data Proc. in Heat and Fluid Flow, London, 1992.
2. R.G. Racca and J.M. Dewey, “A Method for Automatic Particle Tracking in a Three-Dimensional Flow Field,” Experiments in Fluids, 6, 25–32 (1988).

3. B.B. Miller, M.B. Meyer, and M.D. Bethea, "Stereo Imageing Velocimetry for Microgravity Applications," in *Space Optics 1994: Space Instrumentation and Spacecraft Optics*, T.M. Dewandre, J.J. Schulte-in-den-Baumen, and E. Sein, Editors, Proc. SPIE, 2210, 260–270 (1994).
4. N.A. Malik, Th. Dracos, D. A. Papantoniou, "Particle Tracking Velocimetry in Three-Dimensional Flows, Part II: Particle Tracking," Experiments in Fluids, 15, 279–294 (1993).
5. P.D. Crouser, M.D. Bethea, and F. Merat, "Unattenuated SIV Particle Extraction through Median Filtered, Time-Averaged Background Image Subtraction," Experiments in Fluids, 22, 220–228 (19973).
6. P.P.C. Yip and Y.H. Pao, 1994. A guided evolutionary simulated annealing approach to the quadratic assignment problem, *IEEE Transations on Systems, Man and Cybernetics*, vol. 24, no. 9, pp. 1383–1387.
7. K. Nishino, N. Kasagi, and M. Hirata, "Three-Dimensional Particle Tracking Velocimetry Based on Automated Digital Image Processing," Trans. of the ASME, vol. 111, pp. 384–391.
8. T. Kobayashi, T. Saga, and D. Sekimoto, "Velocity Measurement of Three-Dimensional Flow Around Rotating Parallel Disks by Digital Image Processing," ASME Winter Annual Meeting, San Francisco, 1989.
9. R.J. Adrian, "Scattering Particle Characteristics and their Effect on Pulsed Laser Measurements of Fluid Flow: Speckle Velocimetry vs. Particle Image Velocimetry," Applied Optics, 23, 1690–1691 (1984).
10. K.A. Marko and L. Rimai, "Video Recording and Quantitative Analysis of Seed Particle Track Images in Unsteady Flows," Applied Optics, 24, 1985.
11. A.A. Adamczyk and L. Rimai, "2-Dimensional Particle Tracking Velocimetry (PTV): Technique and Image Processing Algorithms," Experiments in Fluids, 6, 373–386 (1988).

TABLE 1.—VALID (I.E. STEREO-MATCHED) AVERAGE TRACK YIELDS AS A FUNCTION OF PARTICLE DENSITY

Number of particles in suspension, (approx.)	Percent of total area covered by the particles	Nonoptimizing model valid track yield	Optimizing model valid track yield	Percent increase in valid tracks found
90 particles	1.34%	66 tracks (73%)	79 tracks (88%)	20%
130	2.17	86 (64%)	107 (82%)	24
190	3.56	117 (62%)	153 (81%)	31
230	4.57	149 (65%)	199 (86%)	33
270	4.86	175 (65%)	221 (82%)	26

Figure 1.—Cleaned camera image of approximately 200 particles.



a. High Velocity constraints b. Medium Velocity constraints c. Low Velocity constraints

Figure 2.—Dependence of Geometrical Constraints upon velocity.

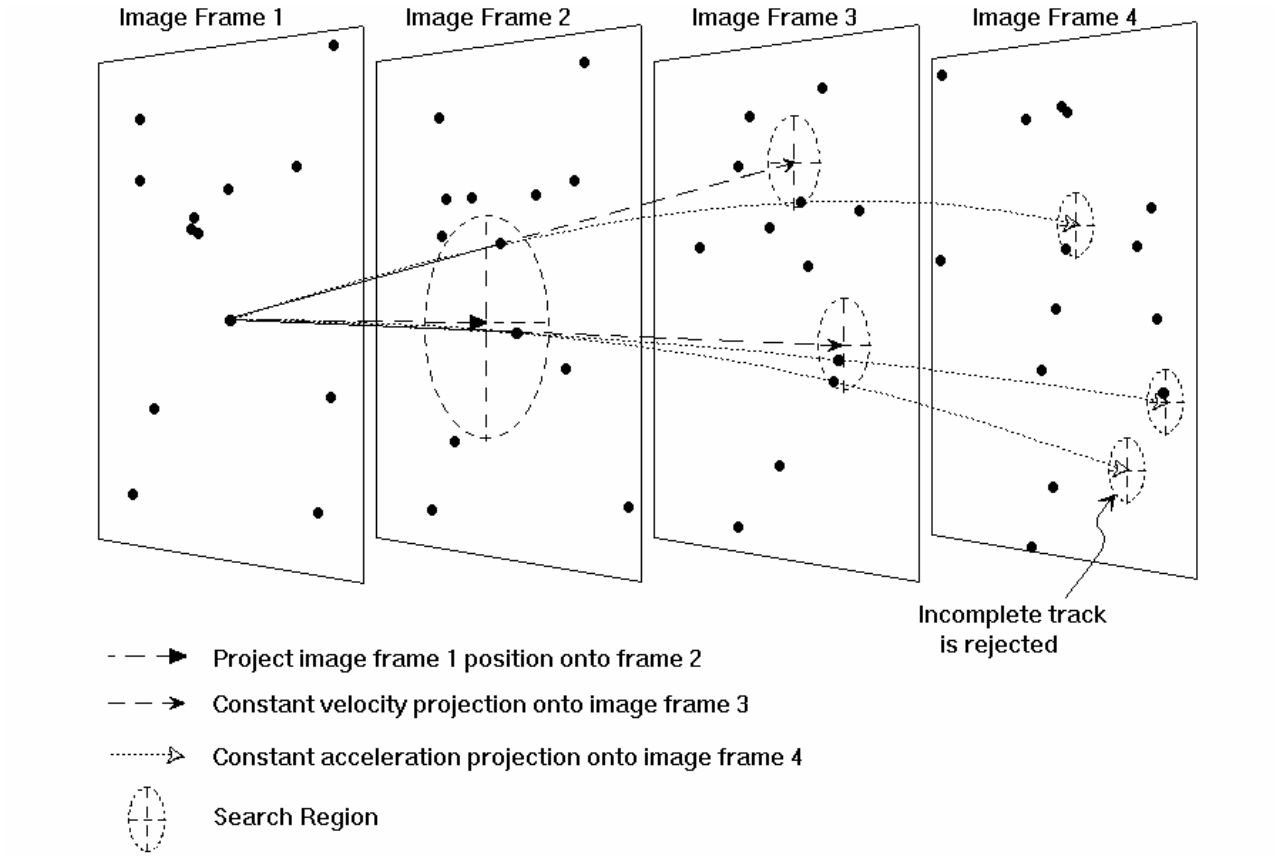


Figure 3.—An example which illustrates the manner in which potentially valid tracks are identified. An empty search region terminates a track fragment. In this example two valid tracks are identified. One of these two track options will ultimately be favored over the other as a result of optimization.

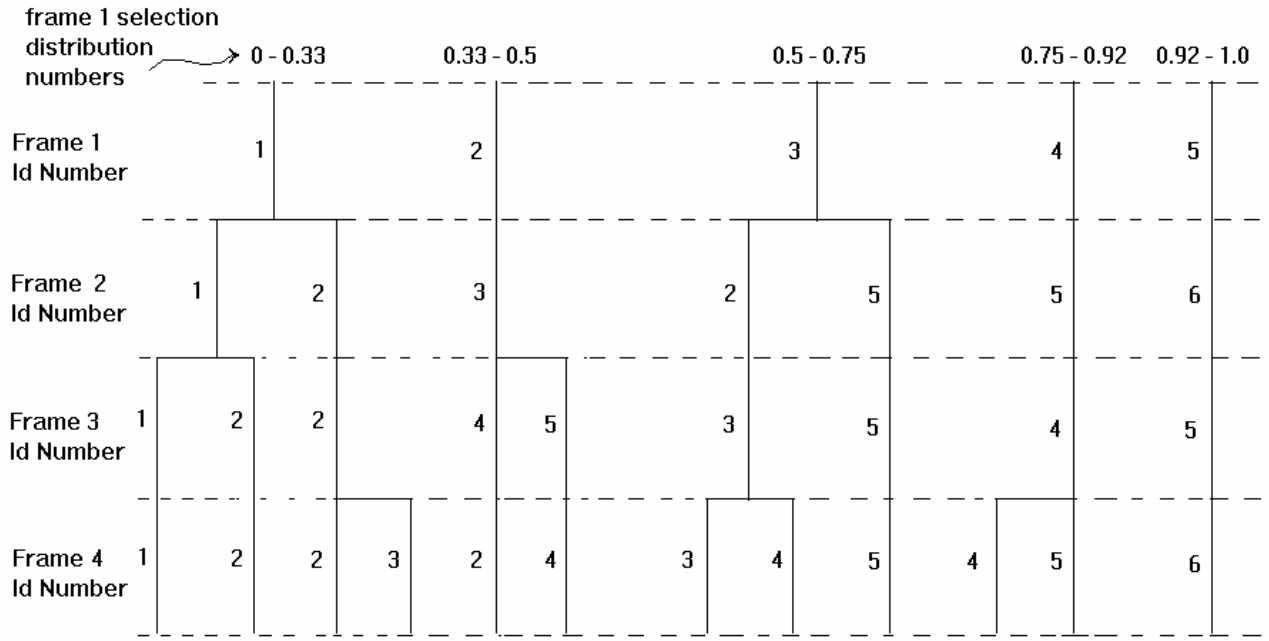


Figure 4.—Depiction of hypothetical valid-track trees. Each particle in each image frame is identified by an id number. Each distinct path from frame 1 through frame 4 represents a distinct track. The frame 1 selection distribution numbers are based upon the ratio of the number of valid tracks in that tree (the number of leaves in the tree) versus the total number of valid tracks for all trees (the total number of valid tracks is 12 in this example).

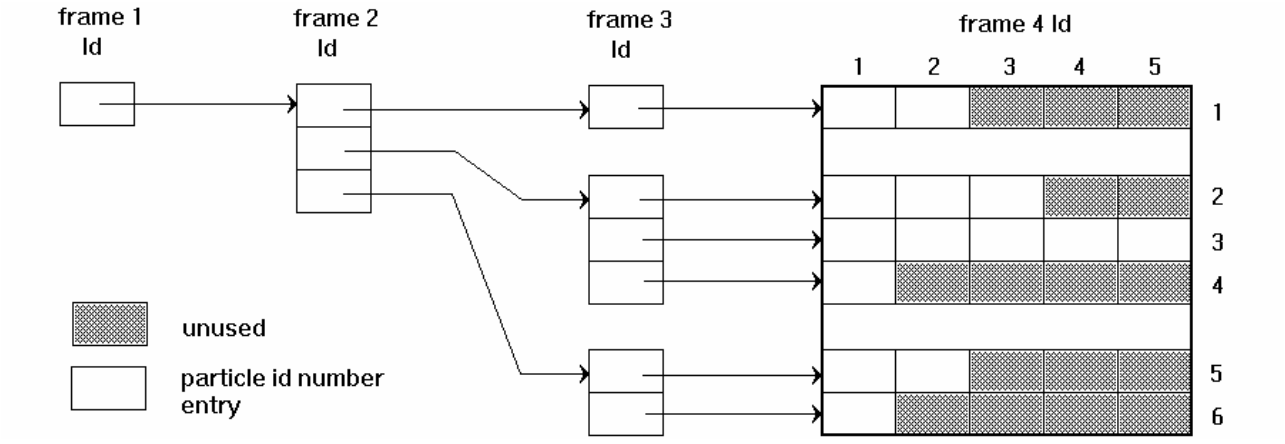


Figure 5.—Whole-track options in a valid-track tree may be represented by the leaves of the tree arranged in a rectangular grid. The best frame 4 option for a given 'track-fragment through frame 3' are placed in column 1 of the grid, the second best in column 2, and so on. Notice that we restrict the maximum number of frame 4 options for each 'track-fragment through frame 3' to a constant number (5 in this case).

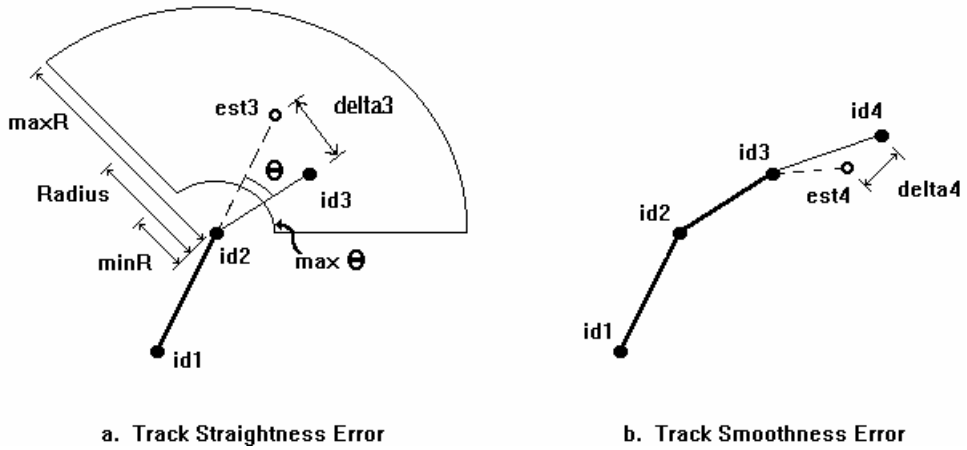


Figure 6.—Track errors.

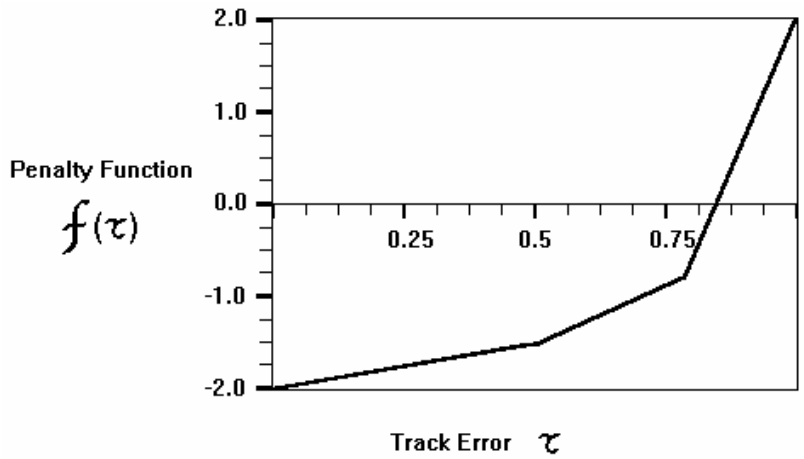


Figure 7.—Mapping from track error to penalty function.

```
while the system error continues to decrease do the following {
        for each parent do the following {
                create children for the parent
                analyze the performance of the children
                determine who will be the new parent
                save the results
                }
        compare families, find the lowest system error, and reward the better
                families with the right to produce more children during the
                next iteration
        update the families by changing over to the new parents
        }
print the results
```

Figure 8.—The main optimization procedure loop.

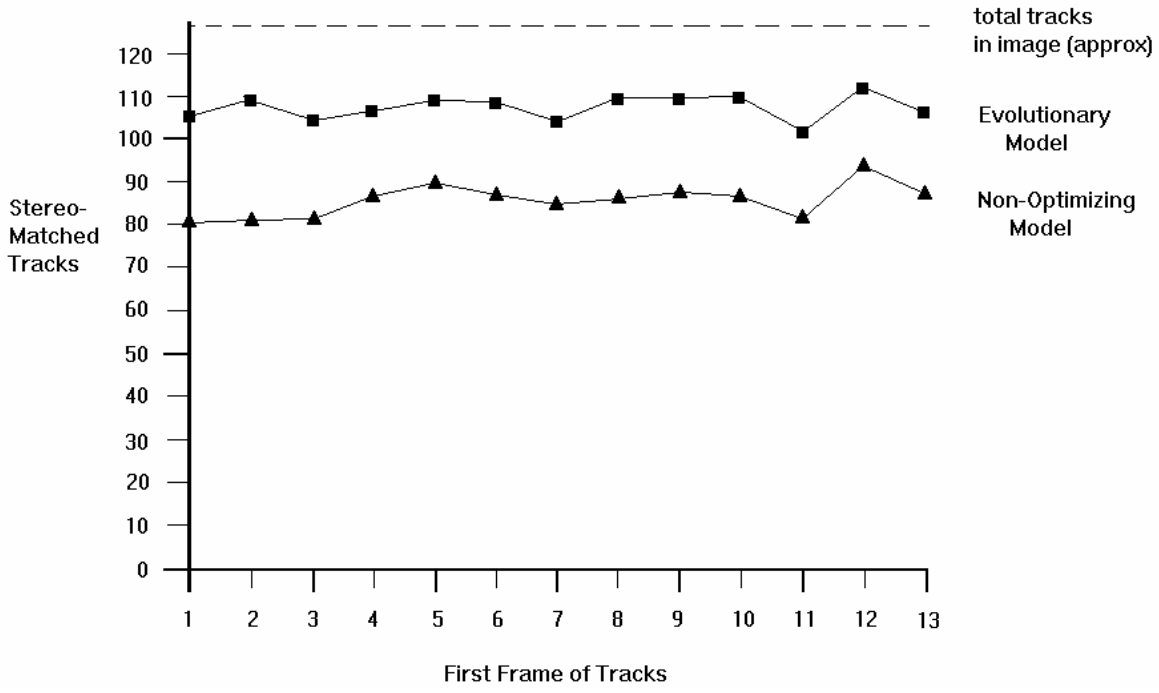


Figure 9.—Comparison of results over 13 + 3 = 16 frames.

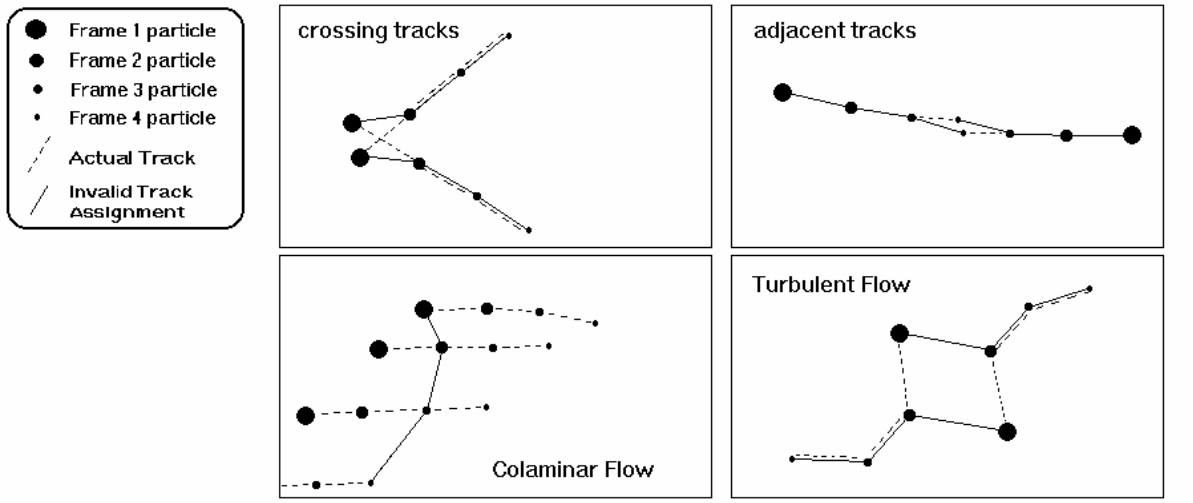


Figure 10.—Examples of failures during optimization.

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-04-2008	Technical Memorandum	

4. TITLE AND SUBTITLE
A Globally Optimal Particle Tracking Technique for Stereo Imaging Velocimetry Experiments

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
McDowell, Mark

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
WBS 561581.02.08.03.16.02

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
John H. Glenn Research Center at Lewis Field
Cleveland, Ohio 44135-3191

8. PERFORMING ORGANIZATION REPORT NUMBER
E-16381

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORS ACRONYM(S)
NASA

11. SPONSORING/MONITORING REPORT NUMBER
NASA/TM-2008-215153

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Categories: 31 and 35
Available electronically at http://gltrs.grc.nasa.gov
This publication is available from the NASA Center for AeroSpace Information, 301-621-0390

13. SUPPLEMENTARY NOTES

14. ABSTRACT
An important phase of any Stereo Imaging Velocimetry experiment is particle tracking. Particle tracking seeks to identify and characterize the motion of individual particles entrained in a fluid or air experiment. We analyze a cylindrical chamber filled with water and seeded with density-matched particles. In every four-frame sequence, we identify a particle track by assigning a unique track label for each camera image. The conventional approach to particle tracking is to use an exhaustive tree-search method utilizing greedy algorithms to reduce search times. However, these types of algorithms are not optimal due to a cascade effect of incorrect decisions upon adjacent tracks. We examine the use of a guided evolutionary neural net with simulated annealing to arrive at a globally optimal assignment of tracks. The net is 'guided' both by the minimization of the search space through the use of prior limiting assumptions about valid tracks and by a strategy which seeks to avoid high-energy intermediate states which can trap the net in a local minimum. A stochastic search algorithm is used in place of back-propagation of error to further reduce the chance of being trapped in an energy well. Global optimization is achieved by minimizing an objective function, which includes both track smoothness and particle-image utilization parameters. In this paper we describe our model and present our experimental results. We compare our results with a nonoptimizing, predictive tracker and obtain an average increase in valid track yield of 27 percent.

15. SUBJECT TERMS
Image analysis; Imaging techniques; Pattern recognition

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email:help@sti.nasa.gov)
U	U	U	UU	23	19b. TELEPHONE NUMBER *(include area code)* 301-621-0390